

Universidad Veracruzana
Licenciatura en Ingeniería de Software

Administración de proyectos de software: Métricas

DCC. Ma. de Lourdes Hernández Rodríguez

Agenda

- Beneficios de medir
- ¿Qué medir?
- Aplicación de métricas
- ¿Cómo se define una métrica?
- Proceso de medición
- Tipos de métricas
- Ejemplos de aplicación
- Técnicas didácticas



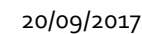
“si no lo puedes medir, no lo puedes mejorar” L.K.

Beneficios de medir

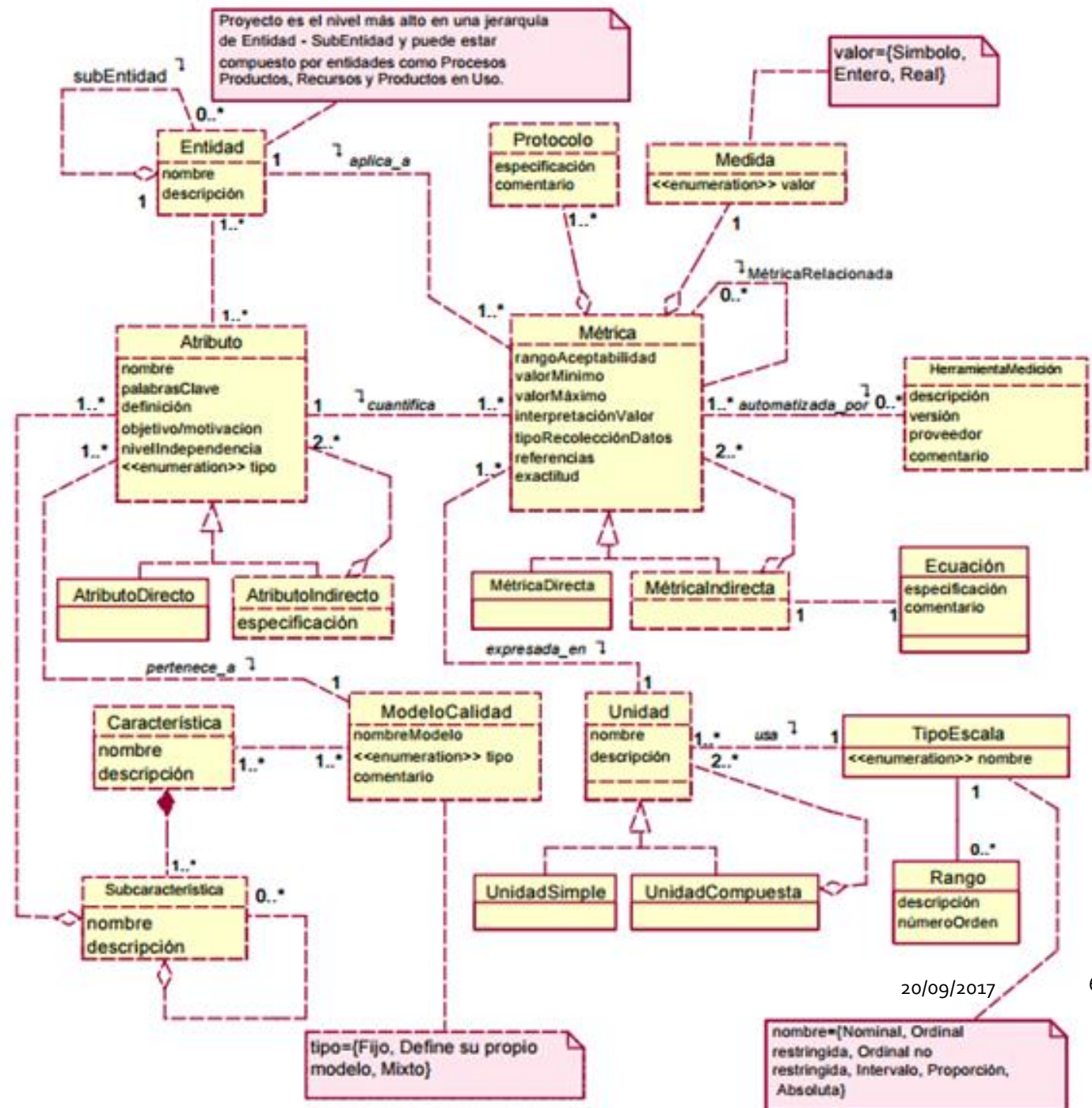
- Planeación y estimación de proyectos
- Administración y seguimiento de proyectos
- Determinación de calidad alineada a objetivos de negocio
- Mejora de procesos, comunicación y herramientas para desarrollo de software





[illegible]

¿Cómo se define una métrica?



Técnicas para definición de métricas

- **Goal-Question-Metrics (GQM):** técnica definida por Basili y Weiss para seleccionar y generar métricas tanto del proceso como de los resultados de un proyecto. Conformado por tres niveles: conceptual (goals), operacional (question), cuantitativo (metrics) .
- **GQ(I)M** Metodología que identifica y define métricas de software que dan soporte a objetivos de negocio de la empresa a nivel de procesos y proyectos. Define el quien, que, cuando, porque y como de los indicadores.
- **PSM:** Practical Software Measurement Basado en experiencias obtenidas por las organizaciones para saber la mejor forma de implementar un programa de medición de software con éxito.
- **Modelos de métricas de software:** ISO 9000 (ISO 9000, ISO 9001, ISO 9003, ISO 9000-3) ISO/IEC 9126. Considerando los atributos: funcionalidad(adaptabilidad, exactitud, interoperabilidad, seguridad), confiabilidad (madurez, tolerancia a fallas, recuperabilidad), usabilidad (comprensibilidad, aprendizaje, operabilidad), eficiencia(comportamiento en función a tiempo, uso de los recursos), mantenimiento (análisis, cambio, estabilidad, prueba).
- **QSOS** Method for qualification and selection of open source software. Propone la evaluación, cuantificación y selección de software open source

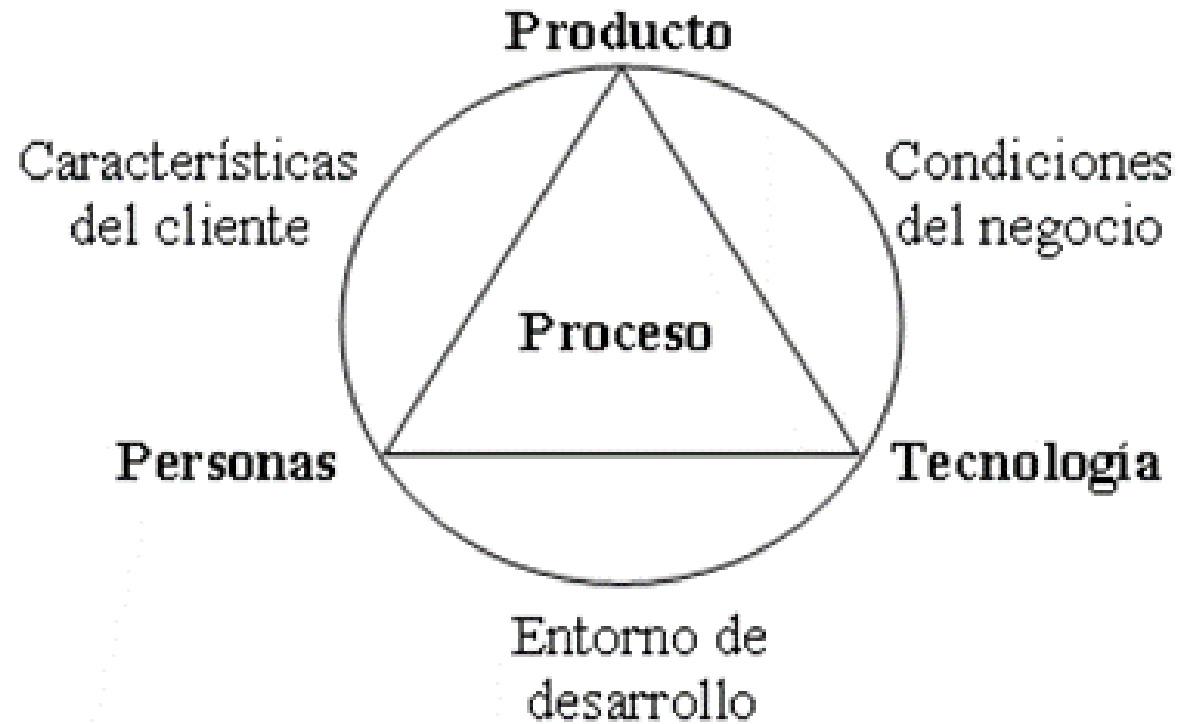


Tipos de métricas

Elemento al que aplican	Aspectos que atienden
Modelo de análisis	<ul style="list-style-type: none">• Funcionalidad entregada, tamaño del sistema(funcionalmente hablando),• Especificación de calidad (especificidad y completitud de requerimientos),• Ejemplos: Puntos de función, Puntos de casos de uso
Modelos de diseño	<ul style="list-style-type: none">• Métricas de arquitectura (estructura, complejidad),• Métricas a nivel de componentes (complejidad),• Métricas de interfaz de usuario (usabilidad),• Métricas a nivel de clases (interfaces, acoplamiento, jerarquía, colaboración)
Código fuente	<ul style="list-style-type: none">• Métricas de complejidad (cohesión, acoplamiento, complejidad ciclomática),• Métricas de longitud (líneas de código, número de entradas, número de salidas)
A nivel de pruebas	<ul style="list-style-type: none">• Métricas de cobertura funcional,• Métricas a nivel de casos de prueba,• Métricas a nivel unidad, integración o sistemas,• Número de defectos permitidos por tipo.

20/09/2017

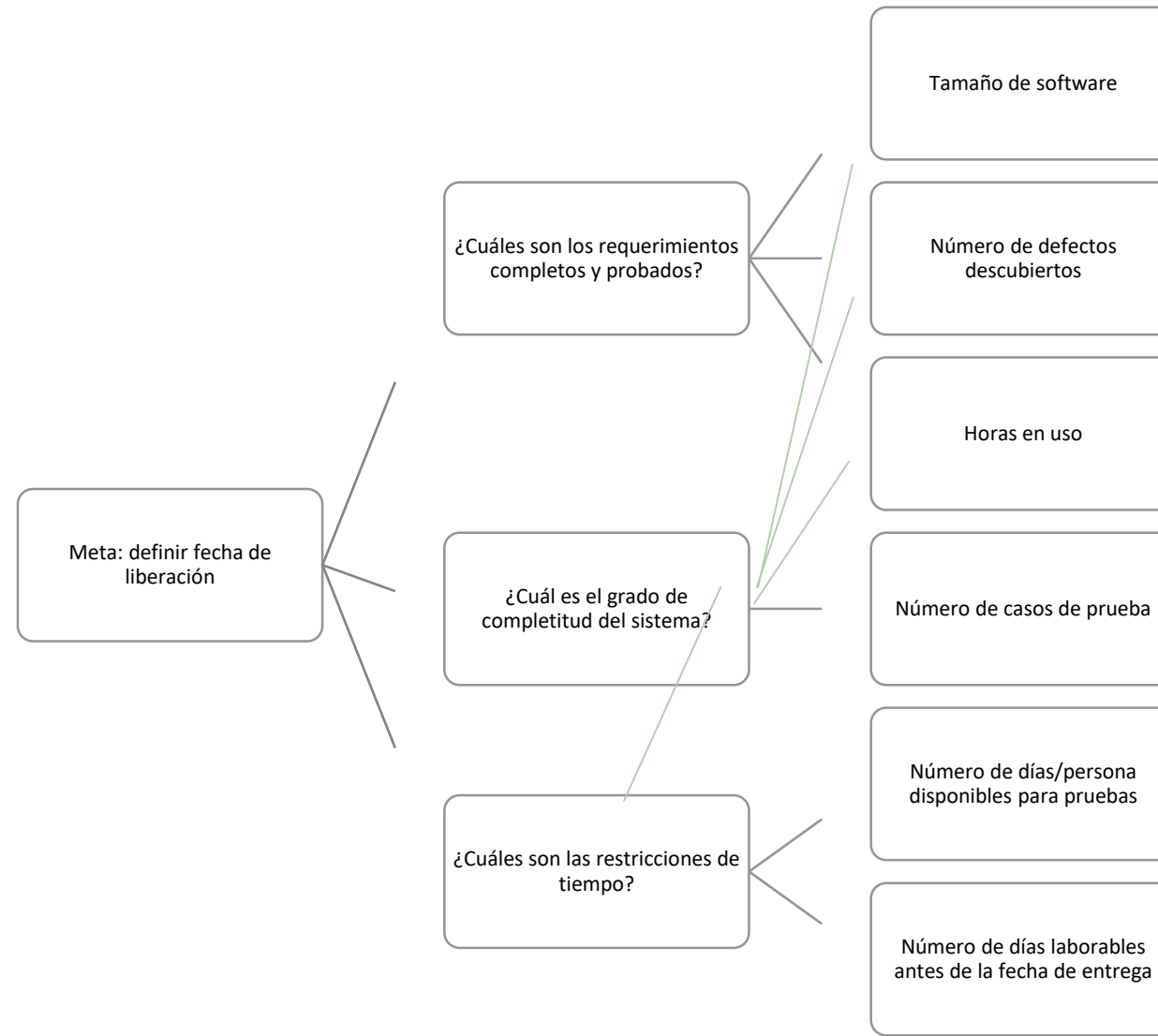
Aspectos que influyen en la calidad del software



Ejemplos de aplicación de métricas

Aspecto	Atributos	Métricas
Proceso de desarrollo	Tiempo invertido	Días calendario, días trabajados
	Puntos de control	Fechas específicas en calendario
	Fase de depuración	Porcentaje total de defectos encontrados
	Fase de entrega o cierre	Porcentaje de tareas completas
Proceso de pruebas	Volumen	Número de pruebas programadas
	Progreso	Número de pruebas ejecutadas
		Número de pruebas pasadas
Mantenimiento	Costo	Pesos por año Horas invertidas por cambio
Sistema	Tamaño	Número de módulos Puntos de función Puntos de casos de uso Memoria requerida
Grupo de trabajo asignado	Tamaño	Número de personal asignado
	Experiencia	Años de experiencia en el dominio de negocio ^{20/09/2017} ₁₁ Años de experiencia en programación

Ejemplo de definición de métricas aplicando GQM



Conclusiones



- Una métrica debe **definirse en función a los objetivos** de negocio que apoya y su interpretación debe ser cuidadosa y objetiva
- Las **métricas** juegan un rol muy **importante** en la **administración** de un proceso de desarrollo de software soportando la toma de decisiones, la predicción o estimación, la gestión y el control. Sin embargo existen situaciones que deben atenderse para asegurar un éxito en su análisis y aplicación como las siguientes:
 - Es primordial **demostrar la utilidad** de una métrica a todos los involucrados en un proceso de desarrollo o mantenimiento de software, considerando el dominio de negocio que se atiende para asegurar su éxito.
 - Se sugiere focalizarse en la medición de un conjunto acotado de atributos de calidad que se complementen y permitan una mejor explicación del **impacto que tienen en un producto** de software
 - Generalmente se tienen **tiempos limitados** para recolección de datos e interpretación de una métrica, de ahí la importancia de utilizar **herramientas** de soporte.

Referencias

- C. Jones, Applied Software Measurement: Global Analysis of Productivity and Quality, vol. 3, Mcgraw-hill, New York, 2008.
- Raymond PL Buse, Thomas Zimmermann, Information needs for software development analytics, in: Proceedings of the 2012 International Conference on Software Engineering, IEEE Press, 2012, pp. 987–996.
- K. Pulford, A. Kuntzmann-Combelles, S. Shirlaw, A Quantitative Approach to Software Management: the AMI Handbook, Addison-Wesley Longman Publishing Co., Inc., 1995.
- R.B. Grady, Practical Software Metrics for Project Management and Process Improvement, Prentice-Hall, Inc., 1992.
- R. S. Pressman. Ingeniería de Software: un enfoque práctico. 5ta ed. Ed. McGraw-Hill. México.2001. pp. 53-75
- Olsina, L., Bertoa, M. F., Lafuente, G., Martín, M. A., Matrib, M., & Vallecillo, A. (2002, November). Un Marco Conceptual para la Definición y Explotación de Métricas de Calidad. In JISBD (pp. 189-200).
- Radatz, J., Geraci, A., & Katki, F. (1990). IEEE standard glossary of software engineering terminology. IEEE Std, 610121990(121990), 3.
- Park, R. E., Goethert, W. B., & Florac, W. A. (1996). Goal-Driven Software Measurement. A Guidebook (No. CMU/SEI-96-HB-002). Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst.



¡Gracias por su atención!